**Public Transport manuscript No.**
(will be inserted by the editor)

# A State-of-the-art Realization of Cyclic Railway Timetable Computation

**Michael Kümmling · Peter Großmann · Karl Nachtigall · Jens Opitz · Reyk Weiß**

Received: date / Accepted: date

**Abstract** We describe the Periodic Event Scheduling Problem (PESP) based on periodic event networks and extend it by symmetry constraints. The modeling power of the PESP is discussed for automatic calculation of feasible periodic railway timetables. Including the described extensions, complete modeling of integrated regular-interval timetables is possible. Encoding PESP to propositional logic enables the usage of efficient SAT solvers for solving PESP. However optimizing timetables by linear programming is possible too. As almost all real-world timetable problems are heavily overconstrained, methods for automatic resolving of conflicts are described. Since there is still a lack of efficient conflict resolving algorithms for large-scale intermeshed railway networks, we introduce several strategies for efficiently resolving conflicts and intelligently decomposing timetable problems and discuss the trade-off between computation time and reduction of the solution space. These strategies allow quick adaption to small changes as well. The described methods are implemented in the software system TAKT. We apply the enhanced TAKT to a timetabling study and present some key figures and results.

**Keywords** timetable optimization · PESP · timetabling · SAT

M. Kümmling · P. Großmann · K. Nachtigall · J. Opitz · R. Weiß
Institute of Logistics and Aviation, Technische Universität Dresden, 01062 Dresden, Germany
E-mail: michael.kuemmling@tu-dresden.de

P. Großmann
E-mail: peter.grossmann@tu-dresden.de

K. Nachtigall
E-mail: karl.nachtigall@tu-dresden.de

J. Opitz
E-mail: jens.opitz@tu-dresden.de

R. Weiß
E-mail: reyk.weiss@tu-dresden.de

## 1 Introduction

Today, planning and disposition of railway transport in Germany is often manual labor – computers only hold, manage and visualize data. These labor-intensive processes lead to mostly evolutionary timetabling, such that every year only the necessary changes are done to fit the timetable to changed infrastructure or a changed network. Using state-of-the-art techniques, it is nearly impossible to build only one fully optimized railway timetable from scratch for large and intermeshed railway networks like the German one. As this obviously does not tap the railway's full potential, diverse high-quality timetable variants have also a key role in design of tomorrow's railway infrastructure and in reliable stability and capacity analysis leading to efficient and sound railway networks.

Since railways are a quite long-standing business, they have grown large bundles of complicated operational rules and versatile constraints. Hence, a model is needed which covers a wide range of possible constraints.

At the Chair of Traffic Flow Sciences at TU Dresden, the software system TAKT has been developed for several years in close collaboration with DB Netz AG. It tackles exactly these kinds of issues and solves them by state-of-the-art operations research techniques. The connection and interaction between the different approaches will be subject of this work. The specific use case TAKT was developed for is long-term timetabling studies. The scope of these studies is to evaluate both different operational concepts (different routes, stopping patterns, vehicles and so on) and different infrastructure states. Even at this long-ranging sight, infrastructure operators, transport companies, regional and national authorities impose many requirements resulting in a high number of constraints. Therefore, conflict resolution plays a major role. Timetable optimization based on predicted passenger demand is a possible post process following conflict resolution.

In Section 2 we give the preliminaries for periodic event scheduling. After showing the possibilities for solving timetabling instances in Section 3, we introduce in Section 4 the state-of-the-art conflict resolving of infeasible instances. After presenting our computational results in Section 5, we conclude the work in Section 6 and give a further scientific outlook.

## 2 Periodic Event Scheduling Problem

In the last 15 years, the Periodic Event Scheduling Problem (PESP) has been established as one of the most suitable problem formulations for periodic timetabling. It is introduced by Serafini and Ukovich [16]. The related periodic event network (PEN) permits flexible representation of almost all periodic timetable's constraints. For instance, PESP and its application to railway timetabling are discussed in detail by Nachtigall and Opitz [13, 14].

The train network, which is the base of the timetabling problem, contains periodic trains $\mathscr{L}$ running on a railway network with stations $\mathscr{S}$. Each train $L \in \mathscr{L}$ serves a specified sequence of Stations $S \in \mathscr{S}$. All constraints are modeled into a PEN. Its nodes $v \in \mathscr{V}$ represent arrival events $(L, arr, S) \in \mathscr{V}$ and departure events $(L, dep, S) \in$

$\mathscr{V}$. The timetable $\mathbf{T} \in \mathbb{Z}^{|\mathscr{V}|}$ assigns to every event $v \in \mathscr{V}$ a potential $T_v \in \mathbb{Z}$, $0 \leq T_v < t_T$. In a periodic timetable with period $t_T \in \mathbb{N}^+$ the event happens periodically at all times $T_v + zt_T$, $z \in \mathbb{Z}$.

The network's arcs $a \in \mathscr{A} \colon i \to j$ are basically time consuming processes. All arcs' processing times are constrained by lower bounds $t_{min,a}$ and upper bounds $t_{max,a}$. This range is also written as $[t_{min,a}, t_{max,a}]_{t_T}$. A timetable $\mathbf{T}$ is considered valid if and only if

$$\forall a \in \mathscr{A} \colon \exists z_a \in \mathbb{Z} \colon t_{min,a} \leq T_j - T_i - z_a t_T \leq t_{max,a}. \tag{1}$$

The *lower slack* $y_a$ is the deviance of the actual processing time from the lower bound such that

$$0 \leq y_a = T_j - T_i - z_a t_T - t_{min,a} < t_T. \tag{2}$$

The PESP is the decision problem, whether there exists any valid timetable for a given PEN $\mathscr{N} = (\mathscr{V}, \mathscr{A}, t_T)$. For feasible problems, a timetable can be calculated.

This universal problem allows the modeling of running times, dwell times, headways and transfer times. For instance, trains are encoded as alternating sequences of running activities $(L, dep, S) \to (L, arr, S')$ and stops $(L, arr, S) \to (L, dep, S)$. Headways between different trains include both safe headways representing the permitted minimum headway and also evenly distributed intervals between different trains running partly on the same railway line. Transfer times include several of different requirements: vehicle transfers, staff transfers and passenger transfers.

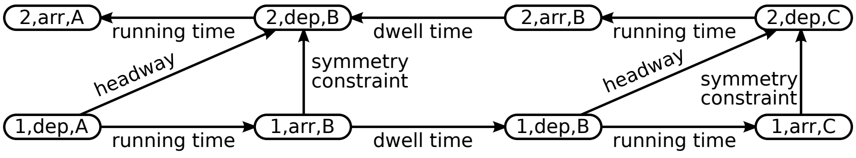

**Fig. 1** An event network of two trains on a single track line with stations A, B and C

Likewise, symmetry is a common requirement in periodic timetabling [17]. A train and its associated returning train are considered to be symmetric, if their arrival and departure times are aligned symmetrically along symmetry axis in time (see Eq. (3)), which is called symmetry minute $s$ and is equal in the whole network. Hence, the trains meet each other at point of time $s$. The symmetric timetables' advantage is the satisfied symmetric property of transfers. Thus, all transfers automatically are fulfilled in both directions. Previous works like [10] and [14] only consider ideal symmetry without any deviation. In practical applications, this is hardly ever accomplishable as even running times are often asymmetric due to slopes or direction dependent maximum speeds. So we introduce a symmetry deviation to model this requirement.

This constraint is special, as it cannot be modeled by Eq. (1) (even without symmetry deviation) as it was proven by Liebchen [10]. Subsequently, the problem has to be extended by additional constraints $a \in \mathscr{A}_S \colon i \to j$, where $i$ is the arrival event of one train and $j$ the departure event of the associated returning train. A certain maximum absolute deviation from symmetry $d_a \in \mathbb{N}$ is permitted. The actual symmetry deviation is denoted as symmetry slack $y_a \in \mathbb{Z}$. Applying the permitted slack (5) to formulation of symmetry axis (3) results in an inequation (6) quite similar to (1).

$$T_j - (s + y_a) - z_a t_T = (s + y_a) - T_i \tag{3}$$

$$T_i + T_j - z_a t_T = 2s + 2y_a \tag{4}$$

$$-d_a \leq y_a \leq d_a \tag{5}$$

$$2s - 2d_a \leq T_i + T_j - z_a t_T \leq 2s + 2d_a \tag{6}$$

Thus, in extension of requirement (1) a timetable is only considered valid if it holds as well:

$$\forall a \in \mathscr{A}_S \colon \exists z_a \in \mathbb{Z} \colon 2s - 2d_a \leq T_i + T_j - z_a t_T \leq 2s + 2d_a \tag{7}$$

The PESP extended by symmetry constraints allows fully modeling Integrated regular-interval timetables (IRIT) [14]. IRIT is a ideal concept of timetabling (refer to [9]) established in Germany and several other European countries. Figure 1 shows a simple exemplifying PESP network with symmetry constraints.

## 3 Solving PESP in TAKT

PESP is proven to be NP-complete [16]. Hence, solving real-world PESP instances is a challenging task [14]. The currently most efficient approach solving PESP is conducted by using state-of-the-art SAT solvers [4]. SAT is the boolean satisfiability problem determining if there exists any interpretation satisfying a given propositional formula. SAT is likewise NP-complete [3], yet, for SAT very efficient solvers exist [11]. It was shown that SAT solvers outperform all previously known approaches for solving PESP despite the additional time needed for encoding and decoding SAT instances [6].

Propositional logic uses boolean variables $q \in \mathscr{R}$. Literals $L$ are either variables $q$ or their negation $\neg q$. Clauses are disjunctions of literals $c = \bigvee_i L_i$. Propositional formulas in conjunctive normal form (CNF) are conjunctions of clauses $\mathscr{F} = \bigwedge_j c_j$. An interpretation $J$ assigns to every variable either the value true or false, denoted as $t$ or $f$, respectively. A formula $\mathscr{F}$ is satisfiable ($\mathscr{F}^J = t$) if and only if there exists a $J$ such that all clauses contain at least one literal assigned to $t$ under $J$. An interpretation $J$ is called model (denoted $J \models \mathscr{F}$), if and only if $\mathscr{F}^J = t$.

The PEN's constraints are encoded to propositional formulas using order encoding which is introduced for general finite ordered domains [18]. All potentials $T_v$ are encoded to $t_T - 1$ boolean variables $q_{n,i}$, whereas $q_{v,i} = t$ represents $T_v \leq i$ and thus, $q_{v,i} = f$ represents $T_v > i$. The function *enc* maps the set of events $\mathscr{V}$ to a propositional formula in CNF, such that this ordering holds:

$$enc(v) = (\neg q_{v,-1} \wedge q_{v,t_T-1}) \bigwedge_{i \in [0, t_T-1]} (\neg q_{v,i-1} \vee q_{v,i}) \tag{8}$$

This encoding for variables of finite ordered domains is discussed in details by [18]. In order to encode all events' potentials, which will be the decoded schedule afterwards, we define

$$\Omega_{\mathscr{N}} := \bigwedge_{v\in\mathscr{V}} enc(v). \tag{9}$$

After a model $J$ has been found for this formula, extracting the value of the event $v$ by $J$ is done by the function $\xi_v(J)$, where $\xi_v(J) = k$ such that $J \not\models q_{v,k-1}, J \models q_{v,k}$ and $k \in [0,t_T-1]$. $\xi_n$ is well-defined and ensures extracting exactly one value due to the following lemma. For detailed consequent proofs we refer to the literature [4,6].

**Lemma 1** *Let $\mathscr{N} = (\mathscr{V},\mathscr{A},t_T)$ be a PEN, $n \in \mathscr{V}$ be an event and $J$ an interpretation. Then*

*(i)* $J \models enc(v) \Leftrightarrow \exists! k \in [0,t_T-1] : \forall i \in [-1,k-1] : J \not\models q_{v,i},$

*(ii)* $\forall j \in [k,t_T-1] : J \models q_{v,j}.$

*Proof (sketch)* "⇒": Let $J \models enc(n)$. To show:

$$\exists k \in [l,u] : \forall i \in [l,k-1] : J \not\models q_{v,i}, \forall j \in [k,u-1] : J \models q_{v,j} \ \wedge \tag{10}$$

$$\nexists h \in [l,u], h \neq k : \forall i \in [l,h-1] : J \not\models q_{n,i}, \forall j \in [h,u-1] : I \models q_{v,j} \tag{11}$$

Equation (10) is simply shown with mathematical induction.
Equation (11) is simply shown without loss of generality $h > k$ implies $q^J_{v,k} = f$, which is in contradiction to (10).
"⇐": can be simply shown with mathematical induction as in "⇒".

Extracting the schedule $\mathbf{T}$ can be done on a per-element basis from a model $J$ with

$$\forall v \in \mathscr{V} : T_v = \xi_n(J). \tag{12}$$

Subsequently, all constraints can easily be encoded to clauses by excluding for each constraint all invalid combinations of values $(T_i,T_j)$. Excluding each single pair would result in a quadratic number of clauses such that each constraint is encoded soundly [4]. Hence, it would be more efficient to encode not just a single pair but a larger set of pairs which will be rectangles. This approach is described in the following. Figure 2 shows the feasible and infeasible regions for the constraint $[3,5]_{10}$ and all pairs of a particular rectangle, that shall be excluded for event $i \in \mathscr{V}$ and $j \in \mathscr{V}$. We can exclude all infeasible pairs of a rectangle $[i_1,i_2] \times [j_1,j_2]$ that is a subset of the infeasible region by a single clause:

$$enc\_rec([i_1,i_2] \times [j_1,j_2]) = \neg q_{i,i_2} \vee q_{i,i_1-1} \vee \neg q_{j,j_2} \vee q_{j,j_1-1} \tag{13}$$

The following lemma ensures that each pair of the encoded rectangle is not satisfiable with respect to the encoded formula and thus, not part of a feasible schedule.

**Lemma 2** *Let $r = ([i_1,i_2] \times [j_1,j_2])$ be a rectangle in the infeasible region of the constraint $(i,j) \in \mathscr{A}$. Then*

$$J \models enc\_rec(r) \Leftrightarrow (\xi_i(J),\xi_j(J)) \notin r$$

*with $J$ being an interpretation.*

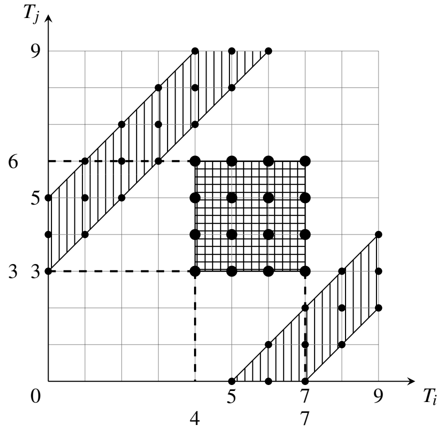

**Fig. 2** Feasible (striped) and infeasible (white) regions for constraint $[3,5]_{10}$ between events $i \in \mathscr{V}$ and $j \in \mathscr{V}$. The gridded square shows an infeasible rectangle.

*Proof* "⇒":

$$J \models enc\_rec(r) \Rightarrow (\neg q_{i,i_2} \vee q_{i,i_1-1} \vee \neg q_{j,j_2} \vee q_{j,j_1-1})^J = t \tag{14}$$

Proof by contradiction: assume $(\xi_i(J), \xi_j(J)) \in r = ([i_1, i_2] \times [j_1, j_2])$. Then

$$i_1 \leq \xi_i(J) \wedge i_2 \geq \xi_i(J) \wedge j_1 \leq \xi_j(J) \wedge j_2 \geq \xi_j(J)$$

which results in

$$\begin{aligned} & q^J_{i,i_2} = t \wedge q^J_{i,i_1-1} = f \wedge \neg q^J_{j,j_2} = t \wedge q^J_{j,j_1-1} = f \\ \Rightarrow\ & (\neg q_{i,i_2} \vee q_{i,i_1-1} \vee \neg q_{j,j_2} \vee q_{j,j_1-1})^J = f \end{aligned}$$

This is a contradiction to (14). Hence,

$$(\xi_i(J), \xi_j(J)) \notin r$$

"⇐": analogous to "⇒".

Excluding all infeasible rectangles of a given constraint $a$ with the given encoding *enc_rec* as of (13) results in the complete and sound encoding for $a$. Applying this to all constraints in the set of edges $\mathscr{A}$ and connecting this with the encoded nodes $\Omega_{\mathscr{N}}$ conjunctively results in the to be encoded propositional formula in CNF. Lemma 1 ensures that we can extract exactly one point in time for each event. Extrapolating Lemma 2 to each constraint ensures the sound and complete encoding for a PEN.

This results in the equivalence of searching for an interpretation $J$ satisfying $\mathscr{F}$ and searching for a valid timetable $\mathbf{T}$ for PEN $\mathscr{N}$. For further reading on encoding PESP to SAT we refer to the literature [6].

An interpretation $J$ satisfying $\mathscr{F}$ can be easily decoded to a timetable $\mathbf{T}$ by reversing the described encoding. Solving PESP by this approach results in one valid timetable, since it is a decision problem. Global timetable optimization can be achieved by minimizing the weighted sum of slacks using integer linear programming (ILP). Lots of different objectives can be modeled by weighting factors, for example sum of journey time for all passengers or the number of needed train sets [10,14]. Yet most implementations of passenger flow based timetable optimization lack proper assignment of passenger flows as described in [7]: Large and complex railway networks require sophisticated traffic assignment methods to achieve a reliable prognosis of traffic streams within the network on the one hand. On the other hand, traffic assignment and timetable optimization are heavily interdepent. This is especially a problem in regional and long-distance train networks with large intervals (120 minutes) between trains. Yet no methods are known to conduct timetable optimization and a sound traffic assignment simultaneously. For a comprehensive survey on further methods for timetable optimization and a wide variety of problem formulations and objectives we refer to [2,19]

## 4 Resolving Conflicts

Although, solving PESP is a challenging task, usually only solving the initially formulated timetable problem is not the scope of work as almost all real timetable problems initially are not satisfiable. This is reasoned by the fact that at first the constraints are arranged idealistically tight, for example dwell times are set to the minimum possible dwell time as this would result in minimal journey times if satisfiable. Therefore, the real task is the identification and resolving of conflicts resulting in a minimally relaxed yet valid timetable.

Let $\mathscr{N} = (\mathscr{V}, \mathscr{A}, t_T)$ be a PEN. A conflict $\mathscr{C} = (\mathscr{V}, \mathscr{Z}, \mathbf{T})$ with $\mathscr{Z} \subseteq \mathscr{A}$ is called local conflict for $\mathscr{N}$ if and only if $\mathscr{C}$ is infeasible and $\mathscr{C}$ gets feasible by removing any constraint in $\mathscr{Z}$. A simple example conflict is outlined in Figure 3. As the event network's constraints are encoded to separate clauses, local conflicts have a counterpart in SAT: A formula $\mathscr{M}$ in CNF is called minimally unsatisfiable subformula (MUS) if and only if $\mathscr{M}$ is unsatisfiable and $\mathscr{M}$ becomes satisfiable by removing any clause $c \in \mathscr{M}$. Likewise, there are highly efficient extractors for finding MUS [15]. Once a MUS is found, it can be decoded back to local conflicts [5].

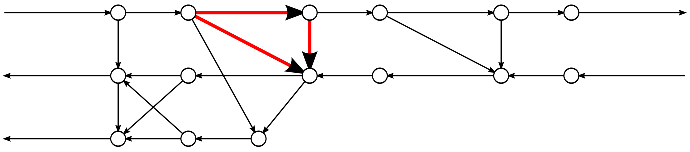

**Fig. 3** Local conflict within a quite small sample event network

Constraints in $\mathscr{Z}$ are relaxed by increasing the upper bound $t_{max,a}$ ($a \in \mathscr{Z}$), whereas symmetry constraints are relaxed by increasing maximum symmetry deviation $d_a$. Several constraints, especially safety headway constraints, but also other constraints by user's request, are prohibited to be relaxed at all. In real world railway PEN, the majority of constraints are unrelaxable headway constraints. Consequently, every conflict at first has to be checked on whether any relaxation of the relaxable arcs could solve the conflict. This is done by removing all relaxable arcs from the network and solving the remaining network. Remaining conflicts are intrinsic conflicts of the railway network's infrastructure and its operating program and thus, have to be manually resolved by modifying the operating program or the infrastructure. The extraction of local conflicts offers a detailed analysis of the bottlenecks [14].

Resolving conflicts involves two steps: Firstly, the network is resolved by a quick heuristic, resulting in far too high relaxations. The simplest heuristic relaxes evenly all relaxable constraints by the same slack until the network is feasible. Afterwards, the relaxations are minimized under preservation of the network's feasibility. Minimization is done by either iterative usage of SAT solvers or direct usage of ILP solvers. The iterative process does not achieve the global minimum, but features much lower calculation times, whereas ILP solvers enable the use of more advanced objective functions.

Despite the impressive speedup achieved by using SAT solvers [6], a lot of timetabling problems are still too vast to be resolved directly in one piece in reasonable time. Therefore, strategies for an intelligent decomposition of the timetabling problem is a necessity. We present two approaches for this problem:

On the one hand, in hierarchical planning, the train network is sub-classified in several levels, for instance long-distance trains, local trains and freight trains as shown in Figure 4. The trains of the highest level are scheduled first and then are left fixed, then the next level is scheduled which is iteratively proceeded. This approach represents the current manual timetabling processes well and easily fits to established paradigms. It reduces calculation time vastly, but it also cuts down the solution space remarkably.

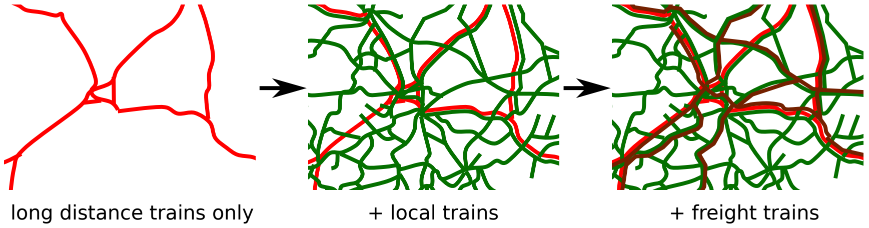

**Fig. 4** Example for hierarchical planning

On the other hand, a new method does not influence the solution space and also results in a considerable reduction of computation time: Some infeasible parts of the timetable network are extracted and resolved separately. Afterwards, all found out relaxations are adopted into the full network, which is then resolved again. Two automatic algorithms for determination of such network parts were developed: local conflict search, as described above and corridor analysis. Corridor analysis extracts the nodes and arcs of a train and its returning train and adds a defined amount of neighboring nodes and arcs. Far more sophisticated algorithms and combined strategies are currently under intensive research. Manual extraction of few heavily crowded and closely intertwined urban networks out of regional or national networks deliver effective results as well.

Both methods allow an intelligent and quick way of incremental scheduling, as they enable to fit small changes into existing timetables easily. Whereas solving whole networks takes hours, it is only a matter of minutes to include changes and to generate valid timetables again. Furthermore, it is possible to evaluate different operating programs and infrastructural states quickly.

## 5 Application and Results

As described in the beginning, the presented algorithms are implemented in the modular timetabling software system TAKT of the Chair of Traffic Flow Science at TU Dresden. The PEN is generated automatically from given input data. This is necessary, as large timetabling problems can consist of up to one million arcs and ten thousands of nodes, which cannot be calculated manually. The program assigns the optimal route on the track layout for each train automatically and calculates the running times within seconds. All minimum headways are calculated individually based on microscopic infrastructure data and the previously calculated running times.

After running time calculation, all running times are considered constant. This is a requirement because the minimum safe headways depend linear but discontinuously on running times [8] and PESP does not allow for dependencies between different arcs. This enables us to reduce PEN size as well by removing arrival arcs and uniting stopping arcs with neighboring arcs (condensation, see [14]). For calculating headways we use commonly established methods. We refer to [12] for a comprehensive overview and to [13] for application in PESP.

This PEN is the universal interface to different PESP solvers, allowing the usage of different solvers like the presented SAT-based approach, but also for instance LP-based solvers. The SAT-based PESP solver uses external SAT solvers. So far, the best overall performance was reached with the Glucose SAT solver [1].

The presented methods are applied to a case study of the German long-distance passenger railway network and the regional trains within the German region south-east (Saxony, Thuringia and Saxony-Anhalt). Firstly, the two networks were solved separately. As the network of regional trains is denser, a particular complex part (Leipzig region) was extracted and the relaxations needed for this part were calculated beforehand. Table 1 shows key figures of these instances.

Calculating a completely conflict-free timetable for the long-distance network takes about 2.5 hours. Determining a valid timetable for the regional trains in the described two iterations took approximately 2 hours each. In the joint network of both long-distance and regional trains, the long-distance trains were fixed to the determined timetable, whereas the regional trains were not fixed, but the relaxations computed before adopted to the event network. Solving the joint network took about 30 minutes. Providing fully correct input data, the program does not need any assis-

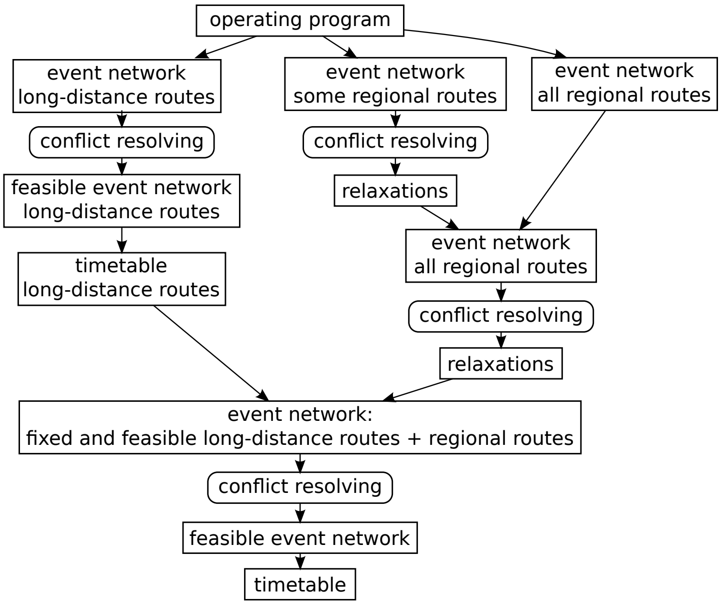

**Fig. 5** Example of hierarchical planning sequence

**Table 1** Key figures of sample instance

| part | trains | stations | nodes | arcs |
|---|---|---|---|---|
| long-distance network | 178 | 198 | 1850 | 14285 |
| Leipzig region | 72 | 126 | 742 | 6337 |
| all regional trains | 426 | 1143 | 4506 | 21461 |
| whole network | 604 | 1329 | 5564 | 31765 |

tance to calculate applicable timetables. Having large amounts of input data, flaws like wrong train routes, missing stops, bogus connections are common. Thanks to the relaxation handling described before, the flaws can be detected and removed iteratively without calculating the timetable from scratch over and over again. The computations were conducted on an Intel Xeon CPU X5450 based server with 16 GB RAM, but used not more than one core and 500 MB RAM. The SAT instances are solved by the SAT solver Glucose version 2.3[1].

Based on PESP a more advanced model for completely automatic calculation of freight train paths along corridors is developed, that allows dynamic track allocations and dynamic selection of suitable speed profiles for freight trains. It outperforms the work of experienced experts even on highly crowded lines by far – the same number of or even more train paths with a better quality are achieved in much lower time [14, 20].

[1] http://www.labri.fr/perso/lsimon/glucose/

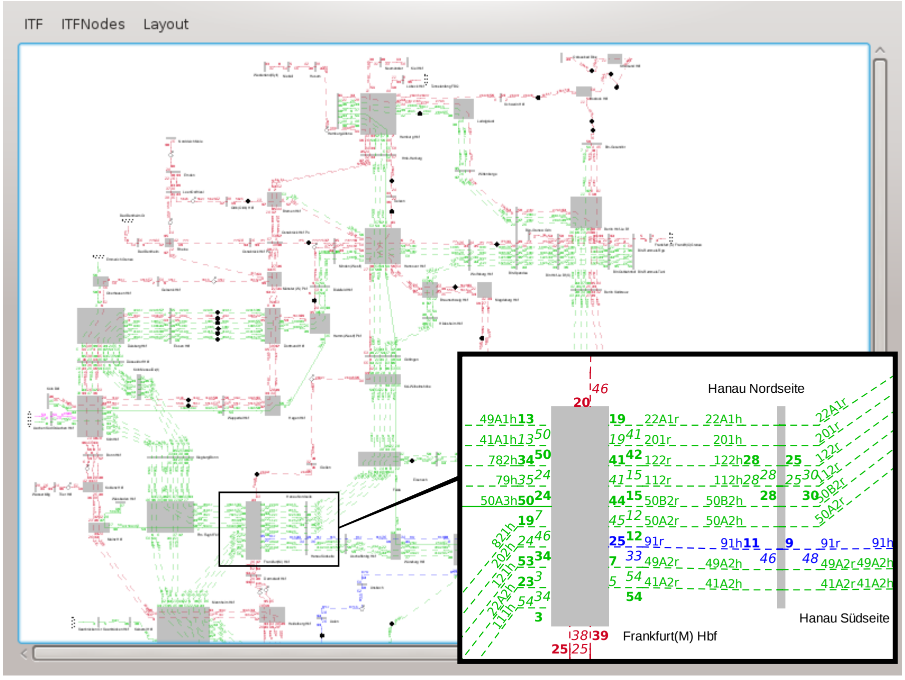

**Fig. 6** Screenshot of timetable visualization in TAKT

Additionally, TAKT features several visualization tools for evaluation of the resulting timetables, one example is displayed in Figure 6.

## 6 Conclusion

As it was shown in this work, SAT-based PESP solving and local conflict search provide a powerful base for fully automatic timetabling. The usage of SAT allows solving larger networks which could not be solved before. Intelligent problem reduction and partition algorithms allow further increments in network size and reductions in calculation time respectively. These improvements permit additional extensions to the PESP model as well. For instance, dynamic track allocation for passenger trains will rise the flexibility of PESP in practical applications further.

The described management of relaxations offers an easy and efficient way for evolving timetables from scratch, which was successfully field-tested on large-scale railway networks. The possibility of fast rescheduling and the variety of realizable constraints opens periodic event scheduling for new fields like capacity research of several infrastructure states or railway traffic management in a completely new manner.

## References

1. Audemard, G., Simon, L.: Glucose 2.3 in the sat 2013 competition. In: Proceedings of SAT Competition 2013 – Solver and Benchmark Descriptions, pp. 42–43. University of Helsinki (2013)
2. Cacchiani, V., Toth, P.: Nominal and robust train timetabling problems. EJOR **219**(3), 727–737 (2012)
3. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the third annual ACM symposium on Theory of computing, STOC '71, pp. 151–158. ACM, New York (1971)
4. Großmann, P.: Polynomial reduction from PESP to SAT. Tech. Rep. 4, TU Dresden (2011)
5. Großmann, P.: Extracting and resolving local conflicts in periodic event networks. Diploma thesis, TU Dresden (2012)
6. Großmann, P., Hölldobler, S., Manthey, N., Nachtigall, K., Opitz, J., Steinke, P.: Solving periodic event scheduling problems with SAT. In: Advanced Research in Applied Artificial Intelligence – 25th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, *Lecture Notes in Computer Science*, vol. 7345, pp. 166–175. Springer (2012)
7. Kümmling, M.: Umlegungsbasierte Fahrplanoptimierung von Personenverkehrstaktfahrplänen. Seminar paper, TU Dresden (2013)
8. Kümmling, M.: Erkennung und Auswertung von signifikanten Konflikten in der Taktfahrplanung. Diploma thesis, TU Dresden (2014)
9. Lichtenegger, M.: Der Taktfahrplan – Abbildung und Konstruktion mit Hilfe der Graphentheorie, Minimierung der Realisierungskosten. phd thesis, Technische Universität Graz (1990)
10. Liebchen, C.: Periodic Timetable Optimization in Public Transport. dissertation.de, Berlin (2006). PhD thesis. TU Berlin
11. Manthey, N., Saptawijaya, A.: Towards improving the resource usage of SAT solvers. In: D.L. Berre (ed.) POS-10, vol. 8, pp. 28–40. Edinburgh (2010)
12. Müller, W.: Betriebstechnische Untersuchung der freien Strecke. Organ für die Fortschritte des Eisenbahnwesens **89**(3), 41–54 (1934)
13. Nachtigall, K.: Periodic network optimization and fixed interval timetable. Habilitation thesis, Universität Hildesheim (1998)
14. Opitz, J.: Automatische Erzeugung und Optimierung von Taktfahrplänen in Schienenverkehrsnetzen. Gabler Verlag, Wiesbaden (2009). PhD thesis. TU Dresden
15. Ryvchin, V., Strichman, O.: Faster extraction of high-level minimal unsatisfiable cores. In: K.A. Sakallah, L. Simon (eds.) SAT, *Lecture Notes in Computer Science*, vol. 6695, pp. 174–187. Springer (2011)
16. Serafini, P., Ukovich, W.: A model for periodic scheduling problems. SIAM Journal on Discrete Mathematics **2**(4), 550–581 (1989)
17. Stähli, S.: Grundfragen der Fahrplangestaltung, 2 edn. Bern (1970)
18. Tanjo, T., Tamura, N., Banbara, M.: A compact and efficient SAT-encoding of finite domain CSP. In: SAT, pp. 375–376 (2011)
19. Törnquist, J.: Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. In: 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05), no. 2 in OpenAccess Series in Informatics (OASIcs) (2006)
20. Weiß, R., Opitz, J., Nachtigall, K.: A novel approach to strategic planning of rail freight transport. In: Operations Research Proceedings 2012, pp. 463–468. Springer, Heidelberg (2014)